Mining Multiple Queries for Image Retrieval: On-the-fly learning of an Object-specific Mid-level Representation

Basura Fernando and Tinne Tuytelaars
KU Leuven, ESAT-PSI, iMinds
Belgium
firstname.lastname@esat.kuleuven.be

Abstract

In this paper we present a new method for object retrieval starting from multiple *query images. The use of multiple queries allows for a more expressive formulation of the query object including, e.g., different viewpoints and/or viewing conditions. This, in turn, leads to more diverse and more accurate retrieval results. When no query images are available to the user, they can easily be retrieved from the internet using a standard image search engine.*

In particular, we propose a new method based on pattern mining. *Using the minimal description length principle, we derive the most suitable set of patterns to describe the query object, with patterns corresponding to local feature configurations. This results in a powerful object-specific mid-level image representation.*

The archive can then be searched efficiently for similar images based on this representation, using a combination of two inverted file systems. Since the patterns already encode local spatial information, good results on several standard image retrieval datasets are obtained even without costly re-ranking based on geometric verification.

1. Introduction

The rapid growth of available image data, thanks to photo sharing sites such as Flickr and Picasa, has raised new challenges in image and video retrieval. Whereas traditional audiovisual archives come with carefully curated metadata, allowing easy access based on a predefined thesaurus, user generated content is hardly annotated – apart from a few, often not very informative, tags. The same holds for many older audiovisual archives that only recently have been digitized. This calls for content-based retrieval methods, that analyze the images directly rather than relying on metadata. In this context, good results have been reported using bag-of-visual-words based methods [3, 5, 16] following the query by example paradigm. Starting from a single query image, these allow efficient retrieval of similar images from the archive building on an inverted file system, similar to the standard text-based retrieval methods using bag-of-words. The result is a ranked list, with images ranked according to their similarity to the given query image.

Figure 1. Multi-query examples with different viewing conditions for query *Eiffel tower* and *London bridge*.

The question then arises: *How to select a good query image?* An object may be pictured from different viewpoints or under different viewing conditions (e.g. different lighting conditions) – see Figure 1. Which images are retrieved then depends, to a large extent, on the selected query image.

To some extent, this can and has been alleviated by developing more robust image representations that are invariant to viewpoint changes and viewing conditions. The image retrieval and computer vision communities have gained considerable progress in this direction with the help of affine invariant representations, domain adaptation techniques, and 'tricks' such as query expansion [3, 5, 6, 18]. Even then, extreme changes in viewing conditions cannot be recovered. Moreover, a single image can only show one aspect of an object (e.g. the front of a car, but not the rear) and may contain substantial amount of image clutter.

Image retrieval starting from *multiple* queries (MQIR) takes a different approach to tackle these problems. By specifying multiple query images, the user can provide more information to the system about the specific object he wants to find without the need to select the best one.

Another interesting application of MQIR is content-based image retrieval based on a textual query. Indeed, by

leveraging the power of internet image search engines such as Google Images, a textual query can be used to retrieve a set of images from the internet. These can then serve as queries for the actual MQIR system. Of course, the set of images returned by the internet search engine may contain errors (i.e., images not representing the object of interest). Therefore, for this type of application, it is important that the MQIR system is *robust to noisy/non-relevant images*.

Given the variety in viewing conditions, different aspects, as well as image clutter, the challenge then is for the system to figure out, purely based on the provided query images, *What exactly is the object of interest?*. Most MQIR methods in the literature (e.g. [2, 12]) do not try to answer this question, yet simply perform a search based on one query image at a time and then fuse the results, or fall back to rather simple schemes to merge the features from the different query images (see also section 2 on related work). In contrast, we rely on an unsupervised pattern mining method that builds an object-specific mid-level representation that best models the query images based on the minimal description length principle (MDL) [10].

To this end, we first create local bag-of-words from a small neighbourhood around each key point to exploit local spatial and structural information. Using these local histograms, we mine the query images to discover a set of *mid-level patterns* that best explain the query images. Our mid-level patterns are compositions of visual words in a specific spatial configuration. They possess local spatial and structural information. We show they are a powerful mid-level representation for image retrieval.

We adapt the recently proposed data mining algorithm KRIMP [25] to discover the visual patterns. KRIMP is based on the minimal description length principle [10]. As such, the standard representation consisting of typically high-dimensional (e.g. 10^6-D) bag-of-words is effectively shrinked into a compact model consisting of just a few hundred patterns.

Our approach has several advantages: (1) It is an unsupervised approach. It does not require any negative images – as opposed to exemplar SVM [2, 14], Joint-SVM [2] or the method presented in [12]. (2) Our method is robust to noisy images as it explores the regularity in the query images using the MDL principle. (3) The resulting KRIMP based representation is compact (only a few hundred patterns). (4) Using our new mid-level representation, relevant images can be retrieved accurately and efficiently, even without costly geometric verification step.

Our major contributions are as follows: (1) We present a multiple query based image retrieval method that systematically outperforms other MQIR methods even without costly spatial verification. (2) We adapt a minimum description length based pattern mining algorithm to discover local structural patterns for specific object retrieval. To the best of our knowledge, we are the first to use MDL based mining for discovering visual patterns. (3) We present a novel query expansion method based on pattern mining which we call pattern based query expansion (**PQE**). (4) We introduce a new dataset to evaluate MQIR methods (See Figure 2).

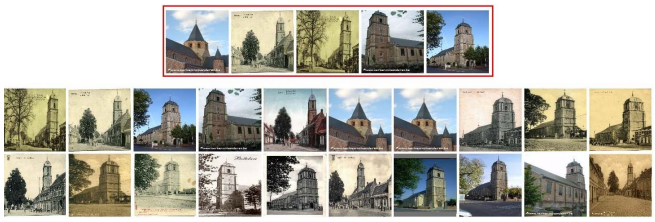

Figure 2. An example query from our new dataset is shown in the first row and retrieved results using our method are shown in the next two rows.

The rest of the paper is organized as follows. First we discuss related work in section 2. Then we present our pattern based multiple query retrieval method in section 3. The experimental evaluation is described in section 4 and the conclusion is given in section 5.

2. Related Work

Single query methods Several methods for large scale image retrieval starting from a single query have been proposed [7, 13, 15, 17, 23]. They almost invariably consist of the following components: bag-of-words creation, indexing, query expansion, and geometric verification.

In *query expansion* [3, 5, 6] (QE), the original query image is replaced with a more representative set of images constructed based on the top ranked images. This, in effect, turns the single query retrieval problem into a multiple query retrieval one.

Geometric verification is typically used to improve the results by re-ranking the top-K retrieved images. This step is linear in the number of top-K images, and therefore needs to be implemented efficiently, e.g. using LO-RANSAC [4].

In addition to these relatively standard components, other improvements have been proposed as well. For instance, Zhang et al. [28] propose to use geometry preserving visual phrases. This method captures geometric information at the feature construction level. However, it is sensitive to local deformations as the structures it discovers are rigid.

Multiple query methods and textual queries In [27] a query specific feature fusion method is proposed. Even though it is not presented as a MQIR method it can be extended to perform MQIR. They combine multiple retrieval sets using a graph based link analysis method and k-reciprocal nearest neighbors [19]. In the work of Liu et al. [12], a textual query based image retrieval method is

presented that uses labeled positive/negative images downloaded from the internet to train a classifier. To handle cross-domain issues they merge domain specific classifiers as proposed in [21]. However, this is only possible if one has sufficient labelled images from each domain. Finally, Arandjelović and Zisserman [2] present several extensions of standard Bag-of-words to perform MQIR. These are described in section 4.1 and serve as baseline for our experimental results. In contrast with all these methods, to the best of our knowledge we are the first to mine query images to discover object-specific patterns to build mid-level features to retrieve images.

Pattern mining In computer vision, pattern mining has been used before mostly for image classification [8, 20, 26] and action classification [9]. In all these cases a set of patterns is mined using the arguably outdated A-priori algorithm [1] or LCM [24], using all the images in the dataset. In contrast we use pattern mining for image retrieval. We mine for patterns specific for an object using small number of query images. We do not mine patterns using the entire dataset. To make sure that the obtained patterns generalize to the image archive, we rely on the minimum description length principle (MDL) [10] to provide us with patterns that are representative of the object. None of the above mining-based methods [8, 9, 20, 26] use MDL to obtain visual patterns. We adapt the recently proposed data mining algorithm KRIMP [25]. Typically KRIMP is used for pattern based classification using a probabilistic model. In contrast we propose to make use of the discovered patterns directly to build mid-level object specific features. To this end, we use the top-k patterns that best represent the query object where the best patterns are selected using a crude MDL principle [10].

3. Pattern Based Image Retrieval

Before we explain our overall process (section 3.3), we first introduce some mining specific notations and terminology (section 3.1) and explain how we construct transactions (section 3.2). We explain pattern mining in detail in section 3.4, our image scoring function in section 3.5, MQBM step in section 3.6 and our query expansion method in section 3.7.

3.1. Notations and terminology

Let I be a set of *items*. In our case, each item is a visual word label (or a label generated for a visual word in a specific configuration–(See section 3.2)). A *transaction* t is a set of items ($t \in PowerSet(I)$), e.g., $t_1 = \{i_1, i_2, i_5, i_9\}$, with each i_j a visual word. Let $D = \{t_1, t_2, t_3 \dots t_n\}$ be what is known as a *database of transactions* that is used to mine patterns from. A *pattern* is a generalization of a set of transactions and consists of a set of items. We denote a pattern by x. Roughly speaking patterns can be considered as subsets of transactions. For example, $x_1 = \{i_1, i_2, i_9\}$ could be a pattern. We say a pattern x is matched/mapped to transaction t if $x \subseteq t$. For example x_1 is matched to t_1. A *model* H is a set of *patterns*[1].

3.2. From visual words to transactions

In our framework, we use root-SIFT descriptors extracted from Hessian-affine regions using the method proposed in [16] along with the root-SIFT descriptor of [3]. A visual vocabulary is created using K-means clustering and each SIFT descriptor extracted from a key point is assigned to one of the visual words (hard assignment). We use local bag-of-words (LBOW) as representation for a local region surrounding a key point. These LBOWs are constructed as histograms over the visual words in the spatial neighbourhood of a detected key point.

Given a key point location and its affine region, we find the spatially k-nearest key points within the affine region to construct a local bag-of-words. The affine region is further split into 4 quadrants aligned with local affine frame, and yet consistent with the upright assumption. This results in a histogram dimension of five times the visual vocabulary size.

Finally, LBOWs are transformed into transactions by considering each non-zero bin as an item.

3.3. Overall Process

Before describing the pattern mining in-detail, we first describe the overall process of our system starting with the offline process, i.e. indexing the archive. Indeed to be applicable in a retrieval scenario, it must be possible to retrieve images from the archive in sub-linear time.

Offline Process : For the archive images, we process the data offline. We create local bag-of-words, transform them into transactions and index them using two inverted file systems (IFS). The first inverted file system, IFS_1, indexes the archive transactions such that the key is a visual word label and the corresponding entry consists of a list of transactions containing this visual word label. The second inverted file system, IFS_2, maps each transaction id to the set of corresponding images. Obviously, only those transactions that actually occur in our database images, are entered in the inverted file systems[2].

Online Process : Given a set of query images and the inverted file systems IFS_1 and IFS_2, we retrieve similar images using the algorithm presented in Algorithm 1. First we extract descriptors for all query images; we assign them to visual words and create transactions as explained in section 3.2. Then we find a set of *consistent* key points (C), i.e.

[1] In section 1.1 of the suppl. material a mining example is presented.

[2] See detailed indexing algorithm in suppl. material (Section 1.2).

key-points that can be found in more than one query image (see section 3.6). Next, we create a database of visual transactions (D) based on the LBOWs describing spatial neighbourhoods around consistent key-points. Then we mine for patterns using the MDL principle and obtain a set of patterns H that best explains the query image visual transactions. For each pattern $x \in H$, we use IFS_1 to find transaction IDs containing this pattern x. Using IFS_2 and the retrieved transaction IDs we then find images that contain this visual pattern x. We repeat this IFS based search for each pattern in the model H. This way we count how many times each pattern occurs in the database images and construct bag-of-patterns. Comparing the bag-of-patterns allows to rank the database images. Finally, this ranked list is refined based on query expansion to obtain the final ranking.[3]

Data: Set of query images, IFS_1, IFS_2
Result: Ranked list
1. Extract features and create local transactions (3.2);
2. Find a set of consistent key-points C (3.6);
3. Create database $D = \{t_i \dots t_n\}$ only from the transactions centering on key-points of C (3.2);
4. $H = \{x_1 \dots x_k\} \leftarrow$ Pattern-Mining(D)(3.4);
5. **for** $x_i \leftarrow 1$ **to** k **do**
   5.1 $T_i = \{t_1 \dots t_m\} \longleftarrow$ Find archive image transactions containing x_i (i.e. $x_i \subseteq t_m$) using IFS_1;
   5.2 $M_i \longleftarrow$ Find set of archive images containing transactions T_i using IFS_2;
end
6. Construct bag-of-patterns from M_i, $i = 1 \dots k$;
7. Ranked list $\longleftarrow$ apply PQE (3.7) and obtain scores;

Algorithm 1: Online image retrieval process.

3.4. Pattern mining

Now that we have created a database of transactions D based on the query images representing the object of interest, we need to explain pattern mining techniques to discover relevant patterns, that can serve as an object-specific mid-level representation for better image retrieval. At this point it's worth noting that a single pattern x only describes a part of the query object. A set of patterns that together describe the query object is called a model denoted by H.

The most widely used pattern mining technique, known as Frequent Itemset Mining (FIM) [1], only discovers patterns that are frequent (as often used in computer vision [9, 20, 26]). With FIM, all closed frequent patterns are used to represent the query object. As a result, a large number of selected patterns describe the same parts of the object and are, in fact, redundant. This model overly represents the data and, consequently, represents the noise too. Hence it is not the most suitable approach for image retrieval. Instead, what we need is a model that best explains the query image data (i.e., the query object) without redundancy.

The KRIMP algorithm[4] [25] provides such solution. It uses the minimal description length (MDL) principle to discover a set of patterns that together best explain the data. The KRIMP algorithm follows the philosophy : *The best models are the ones that compress the data best.*
It exploits the regularity in the query image data to discover relevant patterns to represent the intended object. This approach finds a balance between complexity of the model (number of patterns and complexity of the patterns) and representation of the query data.

Some more terminology Before we explain KRIMP in detail, we introduce a few concepts we will need later on.

A *cover function* is a function of a model H and transaction t, $cover : \{H \times t\} \rightarrow PowerSet(PowerSet(I))$ that indicates which patterns in the model contribute to *cover* the transaction. Intuitively, one could say it selects the model patterns that are used to encode all items in the transaction. Let x, y be patterns such that $x, y \in H$ and t be a transaction ($t \in D$). A cover function has the following three properties: (1) $x \in cover(H,t) \implies x \in H$; (2) if $x, y \in cover(H,t) \implies (x = y)$ or $(x \cap y = \emptyset)$; and (3) $t = \cup_{x \in cover(H,t)} x$. We refer to [25] for more details on how to compute the cover function.

The *usage* of a pattern $x \in H$ by the query image based transactional database D is computed as follows:

$$usage(x|D) = |\{t \in D : x \in cover(H,t)\}|. \quad (1)$$

In words, usage measures how many times the pattern x is used to encode the query images.

KRIMP The KRIMP algorithm then proceeds to select the best model, i.e. the best set of patterns, based on the minimum description length principle, as follows. Given a set of models $\mathbb{H}$, the best model (H^*) is the one that minimizes

$$H^* = argmin_{H \in \mathbb{H}} L(H) + L(D|H) \quad (2)$$

in which $L(H)$ is the length of the model in bits and $L(D|H)$ is the length of the query image data once it is encoded with the model H. First we show how to compute the length of the model $L(H)$:

$$L(H) = \sum_{x \in H} [L(x|H) + L(x|H_{st})] \quad (3)$$

where H_{st} is the standard model consisting of only singleton items. To compute the length of a pattern x given the

[3] See retrieval algorithm in supplementary material (Section 1.3) for more details.

[4] http://www.cs.uu.nl/groups/ADA/krimp/index.php

model, we use Shannon entropy:

$$L(x|H) = -log(P(x|D)), \tag{4}$$

The quantity $P(x|D)$ is computed using the query image information as follows:

$$P(x|D) = \frac{usage(x|D)}{\sum_{\forall y \in H} usage(y|D)}. \tag{5}$$

Next, we look at the second term in eq. 2. The length of the entire query based transactional database D when encoded by the model H, $L(D|H)$, is the summation of all the lengths of transactions in D once encoded by the model H:

$$L(D|H) = \sum_{t \in D} L(t|H). \tag{6}$$

The length of a transaction given the model H and a cover function is then given by

$$L(t|H) = \sum_{x \in cover(H,t)} L(x|H) \tag{7}$$

where $L(x|H)$ is computed as before using equation 4. Note that it can be shown that $L(t|H)$ is equal to $-log(P(t|D))$. Now we can solve the objective function in Equation 2 as described in [25]. By optimizing the above objective function we find the best model H^* that best explains the image query data D with a minimal model complexity. We can use this model directly for image retrieval, following a fully probabilistic approach and ranking the database images based on their probability given the model. Alternatively, we can also consider the set of discovered patterns in the optimal model (H^*) as a new set of mid-level features. Using these patterns we can then construct a histogram (bag-of-patterns) for each image. In practice, we do not use all patterns from H^*, but only the N patterns with smallest length $L(x|H)$. We found experimentally that the fully probabilistic approach is slightly less performing compared to the bag-of-patterns scheme. So we opt to use the latter for all our experiments.

Given a set of multiple query images, we use KRIMP to discover the patterns, followed by Algorithm 1 presented in section 3.3 to build bag-of-patterns and to retrieve similar images. Next we explain the weighting scheme and the similarity function we use to rank images.

3.5. Final score/ranking function

After we have constructed the bag-of-patterns, we use tf-idf weighting. Then we $L2$ normalize the tf-idf weighted bag-of-patterns. To obtain the final score for each database image, we use the square-root histogram intersection similarity [8]. The final score is obtained by summing the square-root histogram intersection similarity over all query images, i.e. for a database image I_d we get

$$score(I_d) = \sum_q \sum_i min(\sqrt{w_d^i}, \sqrt{w_q^i}) \tag{8}$$

where w_d^i is the tf-idf weighted L2 normalized i^{th} bin of the histogram (bag-of-patterns) from database image d and w_q^i is the same for the q^{th} query image.

3.6. Selecting consistent keypoints: Multiple query basic matching (MQBM)

We have now explained most steps of Algorithm 1, except for steps 2 and 7, which we describe in this and next section.

Before we construct the database of transactions, we select consistent key-points. This preprocessing step identifies a set of key points that are consistent across multiple query images. Inconsistent key points, that are found only in a single query image, are filtered out. We call this step *Multiple query basic matching* or **MQBM**.

This is done efficiently using quantized SIFT features, with processing time linear in the number of query images. For this step, we create binary bag-of-words. If a visual word appears in the image, we set it to one, otherwise to zero. By summing all binary-bag-of-words of query images we find visual words that appear at-least among two or more images. Using such visual words, we can find the corresponding consistent key-points. Only transactions centered at consistent key points are selected to construct the database of visual transactions D. This initial multiple query matching step removes unstable and noisy key points, significantly reduces the size of the transactional database and allows to explore larger spatial patterns.

3.7. Pattern based query expansion (PQE)

Query expansion is widely used [3, 5] to obtain better query image representations. For instance, the most widely used average query expansion takes the average of the top-k ranked image histograms. Instead, in our novel pattern based query expansion (**PQE**), we combine the top-k ranked images along with the query images to find a better model (a set of patterns). This can be combined with any pattern mining algorithm. The same objective function used for pattern discovery is used for PQE. For example, for FIM, we select the most frequent set of patterns from both the query images and the top-k ranked images. When KRIMP is used, we re-discover patterns using the query and top-k images using the same MDL objective function (equation 2).

4. Experiments

To evaluate the proposed multiple query image retrieval system we use two mainstream image retrieval datasets: the *Oxford-Buildings-105K [17]* dataset and the Paris-dataset

(6K) [18] with 100K Flicker distractor images (collected from MIRFLICKR-1M [11]) here after referred to as *Paris-105K* dataset. Additionally, we introduce a new challenging dataset to evaluate the performance of multiple query based image retrieval methods. Most of the existing image retrieval datasets are not designed to evaluate multi-query approaches. We have collected old and new images of buildings, statues, churches, castles and other landmarks in Belgium. We call this dataset the Belgium landmarks dataset. Each query consists of five images of a specific place or object from different time periods, viewing conditions and viewpoints. The objective is to retrieve more images of the same place or object using the five query images given. There are eleven queries in the dataset and 5500 images altogether. An example query and the results we obtain using our method is shown in Figure 2.

We use a visual dictionary of one million visual words created using the K-means algorithm. We follow the same experimental setup as in [2]. For both Oxford-105K and Paris-105K datasets we use 5 images per query. Altogether there are 11 multiple query sets for each dataset. We also evaluate all methods using query images obtained using Google image search. We use the top 8 Google images returned for a textual query such as *"Eiffel Tower Paris"* from the Google image search API[5]. To evaluate the performance we use mean average precision (mAP).

4.1. Comparing several multi-query approaches

In this experiment by default we use 10 spatial neighbours of a key-point to construct the LBOWs, KRIMP based pattern discovery and **PQE** for query expansion using the top-5 ranked images (see detailed analysis in section 4.2 for how we selected these parameters). We use only 300 patterns to represent an query object. In Table 1 we compare our method against several multiple query based methods introduced in [2]. These include: i) Joint AVG.: a joint average query method that takes the average of all query histograms; ii) MQ. AVG.: a method that takes the average of scores returned by each single image in the query; iii) MQ. MAX.: a method that takes the maximum instead of the mean of the scores returned by each single image in the query; and iv) Joint SVM: a method that learns a SVM over all query images using a fixed pool of negatives. For Oxford-105K, using dataset queries we report the results for the baselines as reported in [2]. For the other experiments, we used our own implementation, including query expansion. We evaluate the performance with and without using costly spatial verification (SP) step. We did not include the exemplar SVM method included in [2], as it did not perform so well in their experiments on Oxford-105K (mAP of 0.846 with SP). For the spatial verification we use a maximum of 200 top ranked images per query image as in [2].

[5]https://developers.google.com/image-search/

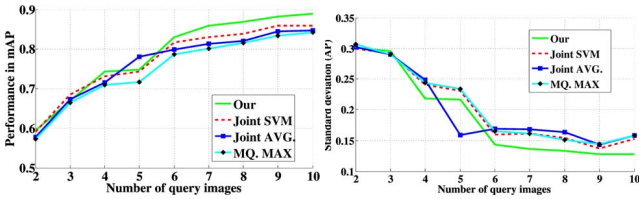

Figure 3. Left: mAP by varying the number of Google images in a query, Right: standard deviation in AP

The KRIMP pattern based method systematically outperforms all other methods. The difference is especially outspoken when not using spatial verification. When Google queries are used, the performance of all methods drops. Yet the ordering of the methods remains the same. On our new challenging dataset, the overall performance of all methods reduces drastically but still the KRIMP mid-level pattern based approach performs best. The single query approach performs very poorly on all datasets. This suggests that, for practical applications and given the ease of collecting query images from the web, multiple queries should be used whenever possible.

In Figure 3 we show how the performance varies with the number of top-K Google images used on Paris-105K dataset. Especially when the number of query images increases, our method outperforms the other methods. Moreover, it also shows less variance than the other methods as the number of images in the query grows.

4.2. Detailed analysis

Next we evaluate some of the design choices we made using Oxford-105K and Paris-105K datasets. For these experiments, we use a smaller visual dictionary of 200K words to keep the computational time low.

Comparing query expansion methods: First, we compare several query expansion methods that can be used in combination with our pattern based approach. In this experiment, we use the FIM pattern mining method to build the query object model and a maximum of 3 neighbors to create LBOWs/transactions. We compare pattern based query expansion (PQE) with the most commonly used average query expansion (AQE) and discriminative query expansion (DQE) introduced in [3]. For DQE we use LibLinear[6] to train the SVM. For all these methods we use the query images along with the top 5 retrieved images for query expansion (in the case of DQE, 5 query images + 5 top ranked images are used as positives and a fixed set of 200 negatives). Results are shown in Figure 4.

From this result we conclude that for our pattern based approach, pattern based query expansion (PQE) is a better choice compared to DQE or AQE. DQE results are disap-

[6]http://www.csie.ntu.edu.tw/cjlin/liblinear/

Method	Oxford-105k						Paris-105k				New Dataset-5K	
	Dataset queries		Dataset queries-our		Google queries		Dataset queries		Google queries		Dataset queries	
	No SV	SV	No SV	SV	No SV	SV	No SV	SV	No SV	SV	No SV	No SV
Single query	0.622 [2]	0.725 [2]	0.616	0.713	0.602	0.668	0.696	0.737	0.539	0.550	0.481	0.501
Joint AVG.	0.886 [2]	0.933 [2]	0.874	0.927	0.743	0.755	0.813	0.892	0.803	0.821	0.738	0.793
MQ. AVG.	0.888 [2]	0.937 [2]	0.876	0.938	0.746	0.755	0.813	0.892	0.769	0.820	0.739	0.794
MQ. Max.	0.826 [2]	0.929 [2]	0.823	0.919	0.673	0.764	0.819	0.889	0.782	0.816	0.752	0.785
Joint SVM	0.886 [2]	0.926 [2]	0.867	0.918	0.750	0.767	0.849	0.888	0.830	0.839	0.675	0.786
Our	**0.947**	**0.944**	**0.947**	**0.944**	**0.767**	**0.790**	**0.907**	**0.913**	**0.865**	**0.869**	**0.791**	**0.797**

Table 1. Comparison of several multiple query approaches with and without spatial verification (SV). For all the methods we compare with in which we don't report results directly from [2], we also include query expansion. For our KRIMP-based method we use pattern based query expansion.

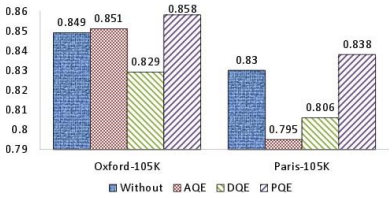

Figure 4. Comparison of query expansion methods.

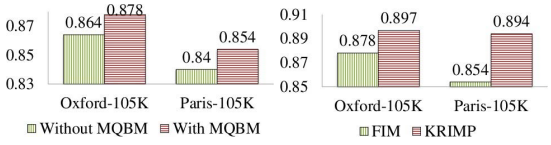

Figure 5. Left: The effect of multiple query basic SIFT matching, Right: Comparison of pattern mining methods: FIM vs. KRIMP.

pointing, may be due to their sensitivity to the selection of negatives. Note that in [6] it is proposed to use the top at most 50 spatially verified results in query expansion and low tf-idf scores provide the negative training data for DQE [3]. Instead, we followed the typical query expansion setup. This could be one of the reason for law performance of DQE.

Effect of initial basic query matching: In this experiment we show the effect of query side basic SIFT matching (MQBM). We construct LBOWs/transactions using a maximum of 10 nearest neighbors and PQE. Results are shown in Figure 5 (left).

Comparison between FIM and KRIMP: Now we compare FIM based pattern mining with KRIMP based pattern mining using all of the above improvements such as PQE and MQBM. In this experiment transactions are created using a maximum of 10 nearest neighbors. Results are shown in Figure 5 (right). Clearly the KRIMP based approach outperforms FIM based approach. In Figure 6 we evaluate the performance of KRIMP pattern based method with varying model size (number of patterns). We sort the patterns by length (i.e. $L(x|H)$) and select the top-k shortest patterns. Even with a small number of patterns such as

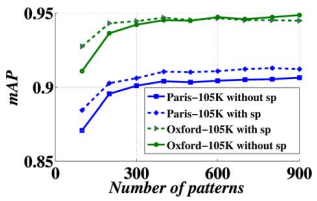

Figure 6. Performance vs. number of patterns using our approach.

100, the KRIMP pattern based model outperforms the baselines of Table 1, with 0.91 mAP without spatial verification and 0.928 with spatial verification on Oxford-105K. At 300 patterns, the curves level off.

Execution times and complexity analysis: Our MQIR method consists of two major steps: (1) processing the query images (SIFT feature extraction, MQBM, transaction creation, pattern mining), and (2) pattern based image retrieval using inverted file systems. The execution time for step (1) only depends on the number of query images. The second step only depends on the size of the image archive. The first step takes about 2-3 seconds for 8 query images while the second step takes less than 0.1 seconds, both measured on a single CPU of 2.6GHz for 250K image archive. In Figure 7, we show how image retrieval time varies with the size of the image archive. The first step of our system can easily be parallelized if multiple CPUs/GPU are available which is a common feature in modern computers. One of the key aspects of our mid-level feature is that they can be indexed for efficient image retrieval. It should be noted here that most of the mid-level feature representations in the literature can't be indexed (e.g. [22]) or at least it is not clear how to do that.

The inverted file systems IFS_1 and IFS_2 take about 1500Mb of ram to index 105K images.[7]

5. Discussion and Conclusion

In this paper we present a new method for image retrieval starting from multiple query images. We learn a

[7]See memory analysis in supplementary material (Section 1.5).

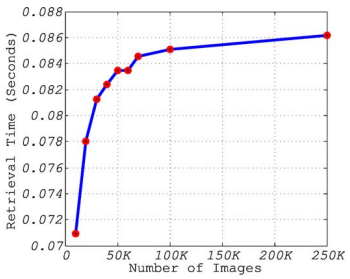

Figure 7. Pattern based retrieval time (excluding the processing time of the query images which is independent of the database size) for 300 patterns by varying the number of images in the image archive.

query object model on-the-fly using a minimal description length principle based pattern mining approach. This way we construct a new mid-level feature representation for the query object. Our mid-level features capture local spatial and structural information so we don't need to rely on costly geometric verification. We also introduced pattern based query expansion which is suitable for pattern based image retrieval methods. We demonstrate excellent results on standard image retrieval benchmarks. Compared to other methods the proposed method shows steady performance improvement as the number of images in a query increases.

We introduced a new challenging dataset to evaluate the performance of multi-query based image retrieval methods. In this dataset even without the costly spatial verification our method again outperforms all others. We show that single query based approaches perform poorly in all datasets. We claim that using multiple queries is a much better choice for image retrieval whenever it is possible to collect several query images. When it is possible to apply multi query based image retrieval, the proposed method seems the most suitable.

Acknowledgements : The authors acknowledge the support of the EC FP7 project AXES and iMinds Impact project Beeldcanon.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[2] R. Arandjelović and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

[3] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[4] O. Chum, J. Matas, and Š. Obdržálek. Enhancing RANSAC by generalized model optimization. In *ACCV*, 2004.

[5] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *CVPR*, 2011.

[6] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[7] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008.

[8] B. Fernando, E. Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In *ECCV*, 2012.

[9] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009.

[10] P. D. Grunwald. *The Minimum Description Length Principle*. THE MIT PRESS, 2007.

[11] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *ACM MIR*, 2008.

[12] Y. Liu, D. Xu, I.-H. Tsang, and J. Luo. Textual query of personal photos facilitated by large-scale web data. *PAMI*, 33:1022 –1036, 2011.

[13] A. Makadia. Feature tracking for wide-baseline image retrieval. In *ECCV*, 2010.

[14] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[16] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[19] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011.

[20] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.

[21] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *NIPS*, 2008.

[22] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[23] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[24] T. Uno, T. Asai, Y. Uchida, and H. Arimura. Lcm: An efficient algorithm for enumerating frequent closed item sets. In *FIMI*, 2003.

[25] J. Vreeken, M. Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.*, 23:169–214, 2011.

[26] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010.

[27] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas. Query specific fusion for image retrieval. In *ECCV*, 2012.

[28] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011.